

FILE ID**PLIDATE

K 13

PPPPPPPPP LL IIIIIII DDDDDDDDD AAAAAAA TTTTTTTTT EEEEEEEEEE
PPPPPPPPP LL IIIIIII DDDDDDDDD AAAAAAA TTTTTTTTT EEEEEEEEEE
PP PP LL II DD DD AA AA TT EE
PP PP LL II DD DD AA AA TT EE
PP PP LL II DD DD AA AA TT EE
PP PP LL II DD DD AA AA TT EE
PPPPPPPPP LL IIIIIII DDDDDDDDD AAAAAAA TTTTTTTTT EEEEEEEEEE
PPPPPPPPP LL IIIIIII DDDDDDDDD AAAAAAA TTTTTTTTT EEEEEEEEEE
PP LL II DD DD AAAAAAAA TT EE
PP LL II DD DD AAAAAAAA TT EE
PP LL II DD DD AA AA TT EE
PP LL II DD DD AA AA TT EE
PP LLLLLLLLLL IIIIIII DDDDDDDDD AAAAAAA TTTTTTTTT EEEEEEEEEE
PP LLLLLLLLLL IIIIIII DDDDDDDDD AAAAAAA TTTTTTTTT EEEEEEEEEE

The diagram illustrates a sequence of binary strings arranged in a grid-like pattern. The strings are as follows:

- Row 1: L
- Row 2: LL
- Row 3: LLL
- Row 4: LLLL
- Row 5: LLLLL
- Row 6: LLLLLL
- Row 7: LLLLLLL
- Row 8: S
- Row 9: SS
- Row 10: SSS
- Row 11: SSSS
- Row 12: SSSSS
- Row 13: SSSSSS
- Row 14: SSSSSSS

(1) 59 plisdate - return date
(1) 89 plistime - return time
(1) 121 subroutines

L 13

```
0000 1 .title plistime_date - pl1 runtime routines for date and time
0000 2 .ident /1-002/
0000 3 :
0000 4 :*****
0000 5 :★
0000 6 :★ COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 7 :★ DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 8 :★ ALL RIGHTS RESERVED.
0000 9 :★
0000 10 :★ THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 11 :★ ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 12 :★ INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 13 :★ COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 14 :★ OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 15 :★ TRANSFERRED.
0000 16 :★
0000 17 :★ THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 18 :★ AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 19 :★ CORPORATION.
0000 20 :★
0000 21 :★ DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 22 :★ SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 23 :★
0000 24 :★
0000 25 :*****
0000 26 :
0000 27 :
0000 28 :++
0000 29 : facility:
0000 30 :
0000 31 : VAX/VMS pl1 runtime library.
0000 32 :
0000 33 : abstract:
0000 34 :
0000 35 : This module contains the pl1 runtime routines that format
0000 36 : date and time
0000 37 :
0000 38 : author:
0000 39 : r. heinen 16-feb-1979
0000 40 : modifications:
0000 41 :
0000 42 :
0000 43 : 1-002 Bill Matthews 29-September-1982
0000 44 :
0000 45 : Invoke macros $defdat and rtshare instead of $defopr and share.
0000 46 :
0000 47 :--
0000 48 :
0000 49 :
0000 50 : external definitions
0000 51 :
0000 52 :
0000 53 :
0000 54 : local data
0000 55 :
0000 56 :
0000 57 : rtshare
```

```

      0000   59      .sbttl plisdate - return date
      0000   60 ;++
      0000   61 ; plisdate - routine to return date
      0000   62 ; functional description:
      0000   63 ; This routine returns the date in YYMMDD format.
      0000   64 ;
      0000   65 ;
      0000   66 ;
      0000   67 ;
      0000   68 ;
      0000   69 ; inputs:
      0000   70 ; r1 = address to return string - char(6)
      0000   71 ;
      0000   72 ; outputs:
      0000   73 ; string is returned.
      0000   74 ;--;
      5E 0E 000C 0000   75 .entry plisdate,^m<r2,r3>
      52 5E D0 0002   76 subl #14,sp          ; allocate buffer
      53 51 D0 0005   77 movl sp,r2
      50 82 3C 0016   78 movl r1,r3          ; save buffer address
      51 51 D4 0019   79 $numtim_s timbuf= (r2) ; get the time
      50 82 3C 0024   80 movzwl (r2)+,r0       ; get year
      51 82 3C 0027   81 clrl r1             ; setup quad word
      51 0039 30 002A   82 ediv #100,r0,r0,r1 ; get remainder from 100
      51 82 3C 002D   83 bsbw cvrt_two_char  ; convert two characters
      51 0033 30 0030   84 movzwl (r2)+,r1       ; get year
      51 82 3C 0033   85 bsbw cvrt_two_char  ; convert to 2 chars
      51 0033 30 0033   86 movzwl (r2)+,r1       ; get day
      51 82 3C 0034   87 bsbw cvrt_two_char  ; convert
      51 0034 04 0033   88 ret                 ; done
      51 50 50 00000064 8F 7B 001B   89 .sbttl plistime - return time
      51 0034 90 ;++;
      51 0034 91 ; plistime - routine to return time
      51 0034 92 ;
      51 0034 93 ; functional description:
      51 0034 94 ;
      51 0034 95 ; This routine returns the time in HHMMSS format.
      51 0034 96 ;
      51 0034 97 ; inputs:
      51 0034 98 ;
      51 0034 99 ; r1 = address to return string - char(6)
      51 0034 100 ;
      51 0034 101 ; outputs:
      51 0034 102 ;
      51 0034 103 ; string is returned.
      51 0034 104 ;--;
      5E 0E 000C 0034 105 .entry plistime,^m<r2,r3>
      52 5E D0 0036 106 subl #14,sp          ; allocate buffer
      53 51 D0 0039 107 movl sp,r2
      50 82 3C 003C 108 movl r1,r3          ; address buffer
      51 0013 30 004A 109 $numtim_s timbuf= (r2) ; save buffer address
      52 06 C0 004D 110 addl #6,r2         ; get the time
      51 82 3C 0050 111 movzwl (r2)+,r1       ; point to time data
      51 82 3C 0053 112 bsbw cvrt_two_char  ; get hour
      51 000D 30 0056 113 movzwl (r2)+,r1       ; convert two characters
      51 82 3C 0059 114 bsbw cvrt_two_char  ; get minute
      51 82 3C 0059 115 movzwl (r2)+,r1       ; convert to 2 chars
      51 82 3C 0059 116

```

51	0007	30	005C	116	bsbw cvrt_two_char	; convert ; convert hundredths ; done
	82	3C	005F	117	movzwl (r2)F,r1	
	0001	30	0062	118	bsbw cvrt_two_char	

ret

0066 121 .sbttl subroutines
0066 122 :++
0066 123 : cvrt_two_char - convert number to two ascii characters
0066 124 :
0066 125 : inputs:
0066 126 :
0066 127 : r1 = number to convert
0066 128 : r3 = address to store data
0066 129 :
0066 130 : outputs:
0066 131 :
0066 132 : r3 = updated address
0066 133 :--
0066 134 cvrt_two_char:
50 51 0A C7 0066 135 divl3 #10,r1,r0 ; get tens digit
83 50 30 81 006A 136 addb3 #^a/0/,r0,(r3)+ ; insert character
50 50 0A C4 006E 137 mull #10,r0 ;
83 51 50 C2 0071 138 subl r0,r1 ; remove tens value
51 30 81 0074 139 addb3 #^a/0/,r1,(r3)+ ; insert ones digit
05 0078 140 rsb ;
0079 141 :
0079 142 .end

PLISTIME_DATE Symbol table

- pl1 runtime routines for date and tim 16-SEP-1984 02:16:50 VAX/VMS Macro V04-00
D 14
6-SEP-1984 11:37:25 [PLIRRTL.SRC]PLIDATE.MAR;1 Page 5
(1)

CVRT TWO_CHAR	00000066	R	01
PLISDATE	00000000	RG	01
PLISTIME	00000034	RG	01
SYSSNUMTIM	*****	GX	01

! Psect synopsis !

PSECT name	Allocation	PSECT No.	Attributes
. ABS . PLISCODE	00000000 (0.) 00000079 (121.)	00 (0.) 01 (1.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	9	00:00:00.06	00:00:00.37
Command processing	65	00:00:00.54	00:00:07.50
Pass 1	67	00:00:00.66	00:00:03.41
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	29	00:00:00.31	00:00:01.38
Symbol table output	0	00:00:00.02	00:00:00.01
Psect synopsis output	2	00:00:00.01	00:00:00.04
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	172	00:00:01.60	00:00:12.98

The working set limit was 750 pages.

2479 bytes (5 pages) of virtual memory were used to buffer the intermediate code.

There were 10 pages of symbol table space allocated to hold 4 non-local and 0 local symbols.

142 source lines were read in Pass 1, producing 14 object records in Pass 2.
7 pages of virtual memory were used to define 1 record.

3 pages of virtual memory were used to define 3 macros.

+-----+
! Macro library statistics !
+-----+

Macro Library name

Macros defined

- \$255\$DUA28:[PLIRTL.OBJ]PLIRTMAC.MLB;1
- \$255\$DUA28:[SYSLIB]STARLET.MLB;2
TOTALS (all libraries)

123

16 GETS were required to define 3 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=TRACEBACK/LIS=LIS\$:PLIDATE/OBJ=OBJ\$:PLIDATE MSRC\$:PLIDATE/UPDATE=(ENHS:PLIDATE)+LIBS:PLIRTMAC/LIB

0307 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

PLICONVRT
LIS

PLICONTRL
LIS

PLIDELTE
LIS

PLIDATA
LIS

PLIDATE
LIS

PLICUTPIC
LIS

PLIENVIR
LIS